IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application: **Nadalin et al.**	§	Group Art Unit: **2132**
	§	
	§	
Serial No.: **09/321,788**	§	Examiner: **Kim, Jung W.**
	§	
Filed: **May 27, 1999**	§	Attorney Docket No.: **AT9-99-081**
	§	
For: **Method for Enabling a Program Written in Untrusted Code to Interact with a Security Subsystem of a Hosting Operating System**	§	Customer No.: **50170**
	§	
	§	
	§	

RESPONSE TO NOTIFICATION OF NON-COMPLIANT APPEAL BRIEF

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

No fees are believed to be required. If, however, any fees are required, I authorize the Commissioner to charge these fees which may be required to IBM Corporation Deposit Account No. 09-0447. No extension of time is believed to be necessary. If, however, an extension of time is required, the extension is requested, and I authorize the Commissioner to charge any fees for this extension to IBM Corporation Deposit Account No. 09-0447.

In response to the Notification of Non-Compliant Appeal Brief dated March 28, 2008, attached is amended section V of the Appeal Brief, i.e. the Summary of Claimed Subject Matter, in which the features of each of the independent claims are identified with references to the specification and figures where appropriate. It is believed that this amended Summary of Claimed Subject Matter meets the requirements set forth in the

Notice. Furthermore, as permitted by the Notification of Non-Compliant Appeal Brief, Appellants are submitting only the amended portion of the Appeal Brief rather than a substitute Amended Appeal Brief. However, should the Board require an Amended Appeal Brief, Appellants respectfully request that the Board contact Appellants' undersigned representative at the telephone number below and one will be provided.

Respectfully submitted,

DATE: April 21, 2008

Stephen J. Walder, Jr.
Reg. No. 41,534
Walder Intellectual Property Law, P.C.
P.O. Box 832745
Richardson, TX 75083
(214) 722-6419
ATTORNEY FOR APPELLANTS

V. Summary of Claimed Subject Matter

With regard to independent claim 11, a method for enabling a program written in untrusted code (e.g., 35 in Figure 1; page 9, lines 2-7) to access a native operating system resource (e.g., 34 in Figure 1) is provided (e.g., page 4, lines 20-23; page 5, lines 3-6). The method comprises having a trusted login service listen on a named pipe (e.g., 78 in Figure 2; page 10, lines 8-13; page 11, lines 3-5; page 14, lines 5-6) for login requests (e.g., page 5, lines 6-8). The method further comprises, responsive to a login request (e.g., 70 in Figure 2; page 10, lines 13-15; page 14, lines 6-7), wherein the login request contains an identifier for a uniquely-named response pipe (e.g., 81 in Figure 3; page 10, lines 8-11; page 11, lines 15-18), having the trusted login service request a native operating system identifier (e.g., page 5, lines 8-10). Moreover, the method comprises returning to the program via the uniquely-named response pipe the native operating system identifier (e.g., 83 and 89 in Figure 3; page 5, lines 10-12; page 11, line 19; page 12, line 1; page 14, lines 13-16). The uniquely-named response pipe and the named pipe are not identical (e.g., service pipe and response pipe; page 10, lines 8-11). In addition, the method comprises, in an authentication framework (e.g., 38, 40, 42, and 44 in Figure 1; page 9, lines 9-13), using the native operating system identifier to create a credential object (e.g., 96 in Figure 2; page 12, lines 21-24) and using the credential object to login to the native operating system to enable the program to access the resource (e.g., 38 in Figure 1; page 5, lines 12-16; page 12, lines 2-5; page 13, line 2).

With regard to independent claim 17, a computer program product in a computer readable medium for enabling a program (e.g., page 18, lines 9-18) written in untrusted code (e.g., 35 in Figure 1; page 9, lines 2-7) to access a native operating system resource (e.g., 34 in Figure 1) is provided (e.g., page 4, lines 20-23; page 5, lines 3-6). The computer program product may comprise means for listening on a named pipe by a trusted login service (e.g., 78 in Figure 2; page 10, lines 8-13; page 11, lines 3-5; page 14, lines 5-6) for login requests (e.g., page 5, lines 6-8) and means responsive to a login request (e.g., 70 in Figure 2; page 10, lines 13-15; page 14, lines 6-7) for requesting a native operating system identifier by the trusted login service (e.g., page 5, lines 8-10). The login request contains an identifier for a uniquely-named response pipe (e.g., 81 in

Figure 3; page 10, lines 8-11; page 11, lines 15-18). The computer program product further comprises means for returning to the program via the uniquely-named response pipe the native operating system identifier (e.g., 83 and 89 in Figure 3; page 5, lines 10-12; page 11, line 19; page 12, line 1; page 14, lines 13-16). The uniquely-named response pipe and the named pipe are not identical (e.g., service pipe and response pipe; page 10, lines 8-11). The computer program product further comprises, in an authentication framework (e.g., 38, 40, 42, and 44 in Figure 1; page 9, lines 9-13), using the native operating system identifier to create a credential object (e.g., 96 in Figure 2; page 12, lines 21-24) and using the credential object to login to the native operating system to enable the program to access the resource (e.g., 38 in Figure 1; page 5, lines 12-16; page 12, lines 2-5; page 13, line 2).

With regard to independent claim 21, an application server (e.g., 12 in Figure 5; page 15, lines 25-26) is provided that comprises a set of programs (e.g., 16a-16n in Figure 5; page 16, lines 12-13) that are supported by a virtual machine (e.g., page 16, lines 2-8) that is supported by a native operating system (e.g. see 12 in Figure 5; page 16, lines 2-8) and a processor (e.g., see 12 in Figure 5; page 16, lines 2-8) running the native operating system providing support for executing the set of programs (e.g., see 12 in Figure 5; page 16, lines 2-8). The application server further comprises means for enabling each program in the set of programs to run in an operating system thread (e.g., page 16, lines 13-19) while impersonating a different native operating system user (e.g., page 16, lines 13-19) in accordance with a token (e.g., 88 in Figure 2, page 12, lines 11-12) that was created during a login operation (e.g., page 12, lines 20-23) in the native operating system (e.g. see 12 in Figure 5; page 16, lines 2-8) and that was associated with a program (e.g., 35 in Figure 1; page 9, lines 2-7) while the program was acting as a named-pipe server to listen for a login response on a named pipe (e.g., 78 in Figure 2; page 10, lines 8-13; page 11, lines 3-5; page 14, lines 5-6) that was uniquely created for a login request (e.g., page 5, lines 6-8) to obtain the token (e.g., 83 and 89 in Figure 3; page 5, lines 10-12; page 11, line 19; page 12, line 1; page 14, lines 13-16). The login request contained an identifier for the named pipe (e.g., 81 in Figure 3; page 10, lines 8-11; page 11, lines 15-18).

Regarding independent claim 24, a method for enabling a program written in untrusted code (e.g., 35 in Figure 1; page 9, lines 2-7) to access in a trusted manner a resource (e.g., 34 in Figure 1) supported on a computing device executing a native operating system (e.g. see 12 in Figure 5; page 16, lines 2-8) is provided (e.g., page 4, lines 20-23; page 5, lines 3-6). The method comprises listening, by a trusted login service in the native operating system (e.g., 78 in Figure 2; page 10, lines 8-13; page 11, lines 3-5; page 14, lines 5-6), for login requests on a named request pipe and generating a login request at the program (e.g., page 5, lines 6-8). The login request contains authentication information (e.g., 70 in Figure 2; page 10, lines 13-15; page 14, lines 6-7) and an identifier for a named response pipe (e.g., 81 in Figure 3; page 10, lines 8-11; page 11, lines 15-18). The named request pipe and the named response pipe are not identical (e.g., service pipe and response pipe; page 10, lines 8-11). The method further comprises, in response to creating the named response pipe by the program, acting as a named-pipe server on the named response pipe by the program (e.g., 78 in Figure 2; page 10, lines 8-13; page 11, lines 3-5; page 14, lines 5-6) and, in response to receiving the login request on the named request pipe (e.g., service pipe; page 10, lines 8-11) at the trusted login service from the program, performing a login operation with the authentication information (e.g., user ID and password; page 10, lines 15-19) by the trusted login service into the native operating system (e.g., 74 in Figure 2). The method further comprises, in response to performing the login operation (e.g., 74 in Figure 2), sending a login response (e.g., 84 in Figure 2) on the named response pipe from the trusted login service to the program (e.g., 84 in Figure 2; page 11, lines 10-13). Moreover, the method comprises, in response to receiving the login response (e.g., 84 in Figure 2) on the named response pipe at the program from the trusted login service, closing the named response pipe (e.g., 91 in Figure 3) such that the named response pipe is uniquely associated with the login request and is not used for additional login requests (e.g., page 12, lines 1-2). The method also comprises, in response to receiving the login response on the named response pipe at the program from the trusted login service, creating a credential object (e.g., 96 in Figure 2; page 12, lines 21-24) by the program using a token generated during the login operation. In addition, the method comprises using the credential object by the program to access the resource within the native

operating system (e.g., 38 in Figure 1; page 5, lines 12-16; page 12, lines 2-5; page 13, line 2).